



# RTAI 3.8 ON UBUNTU(9.10)-LINUX-KERNEL : 2.6.31.8

Manuel Arturo Deza

## **1: Installing Rtai 3.8**

The following Tech Report / Guide is a compendium of instructions necessary for installing RTAI 3.8 on Ubuntu 9.10 using the Kernel 2.6.31.8. These instructions have been uploaded to the author's personal webpage: <http://arturodeza.wikidot.com/data-log>

### **1) Download RTAI 3.8**

<https://www.rtai.org/>

Download and uncompress : RTAI 3-8 on a new folder in Documents called RTAI\_vf.

### **2) Download Kernel 2.6.31.8**

Open a Terminal and type:

```
$ cd /tmp
```

```
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.8.tar.bz2
```

Now as "SuperUser":

```
$ sudo su
```

```
#tar -xjvf linux-2.6.31.8.tar.bz2 -C /usr/src
```

New Kernel 2.6.31.8 is now uncompressed at /usr/src !

NOTE:

Importante : USAR EL KERNEL 2.6.31.8 Y RTAI 3.8 SOLO SI SU ARQUITECTURA DE PROCESADOR ES DE TIPO x86.

USE THE 2.6.31.8 & RTAI 3.8 ONLY IF THE PROCESSOR ARCHITECTURE IS OF TYPE x86.

How do I find out that I'm using an x86 processor architecture?

Open a terminal and type:

```
$ uname -a
```



You should get something like this:

```
Linux username-laptop 2.6.31-14-generic #48-Ubuntu SMP Fri Oct 16 14:04:26 UTC 2009 i686
GNU/Linux
```

Check that before "GNU/Linux" you get "i686". If that's the case then here we go!

### **3) Apply the Patch over the Kernel Code.**

Open a terminal:

```
$ cd /usr/src/linux-2.6.31.8
$ patch -p1 </home/username/Documents/RTAI_vf/rtai-3.8/base/arch/x86/patches/hal-linux-
2.6.31.8-x86-2.4-09.patch
```

NOTE: Replace username with your OWN user name.

### **4) Configure the Kernel with the patch**

(You'll need to get the QT Development package installed)

Log in as root:

```
$ sudo su
#make xconfig
```

Remember to Select/Unselect these options:

Unselect:

- \* Support for large (2TB+) block devices & files.
- \* Symmetric Multi-processing support
- \* Local APIC support on uniprocessors

Select:

- \* Enable loadable module support.
- \* Choose form the Radio buttons your processor type (I have a Core2).

Save Configuration/Settings.

Most mistakes on installing RTAI happen if you misselect some options, or activate and deactivate accidentally some xconfig settings. In case errors appear at Kernel booting time, check the Internet or on other Instruction manuals about possible reasons.



## 5) Compile Kernel with the Patch

#make

#make modules

#make modules\_install

```
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$ make xconfig
CHECK qt
*
* Unable to find the QT3 installation. Please make sure that
* the QT3 development package is correctly installed and
* either install pkg-config or set the QTDIR environment
* variable to the correct location.
*
sed < scripts/kconfig/lkc_proto.h > scripts/kconfig/lkc_defs.h 's/P(\[^\,]*\),,.*
/define \1 (\[^\,]*\)/'
HOSTCC scripts/kconfig/kconfig_load.o
make[1]: *** No rule to make target 'scripts/kconfig/.tmp_qtcheck', needed by 's
cripts/kconfig/qconf.o'. Stop.
make: *** [xconfig] Error 2
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$ make xconfig
CHECK qt
/usr/bin/moc -i scripts/kconfig/qconf.h -o scripts/kconfig/qconf.moc
HOSTCXX scripts/kconfig/qconf.o
HOSTLD scripts/kconfig/qconf
scripts/kconfig/qconf arch/x86/Kconfig
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$ make clean
manueldeza@manueldeza-laptop:~/Documents/RTAI/linux-2.6.32.2$
```

## 6) Install the Kernel

#make install

You'll find 3 new GRUB files:

System.map-2.6.31.8

config-2.6.31.8

vmlinuz-2.6.31.8

## 7) Create an initrd "Image file" for the booting process.

#update-grub

## 8) Test and reboot your computer.



#reboot

When you reboot your computer, count to 20 (20 seconds) you'll probably hear a soft "click", **immediately after GRUB initializes**. I think this is necessary for the Kernel to not bug itself. I can't really figure the real reason why, but it's best to follow this recommendation. Select your new kernel 2.6.31.8 and wait for the system to initialize.

If the system initializes without bugs, then reboot again to your older version kernel.

## 9) Configure RTAI

Open a terminal in your preconfigured kernel:

Replace username with YOUR user name.

```
$ cd /home/username/Documents/RTAI_vf
```

```
$ mkdir rtai-build
```

```
$ cd rtai-build
```

```
$ make -f /home/username/Documents/RTAI_vf/rtai-3.8/makefile xconfig
```

Go to (Browse):

General-> Linux Source Tree and set (double click on the current address) the address to:  
/usr/src/linux-2.6.31.8

## 10) Make RTAI

Super user mode:

```
$ sudo su
```

```
#make
```

## 11) Install RTAI

```
#make install
```

## 12) Todo Listo ! Hora de probar RTAI!

#reboot

Reboot your computer and remember the 20 second rule, to boot in your RTAI configured Kernel.

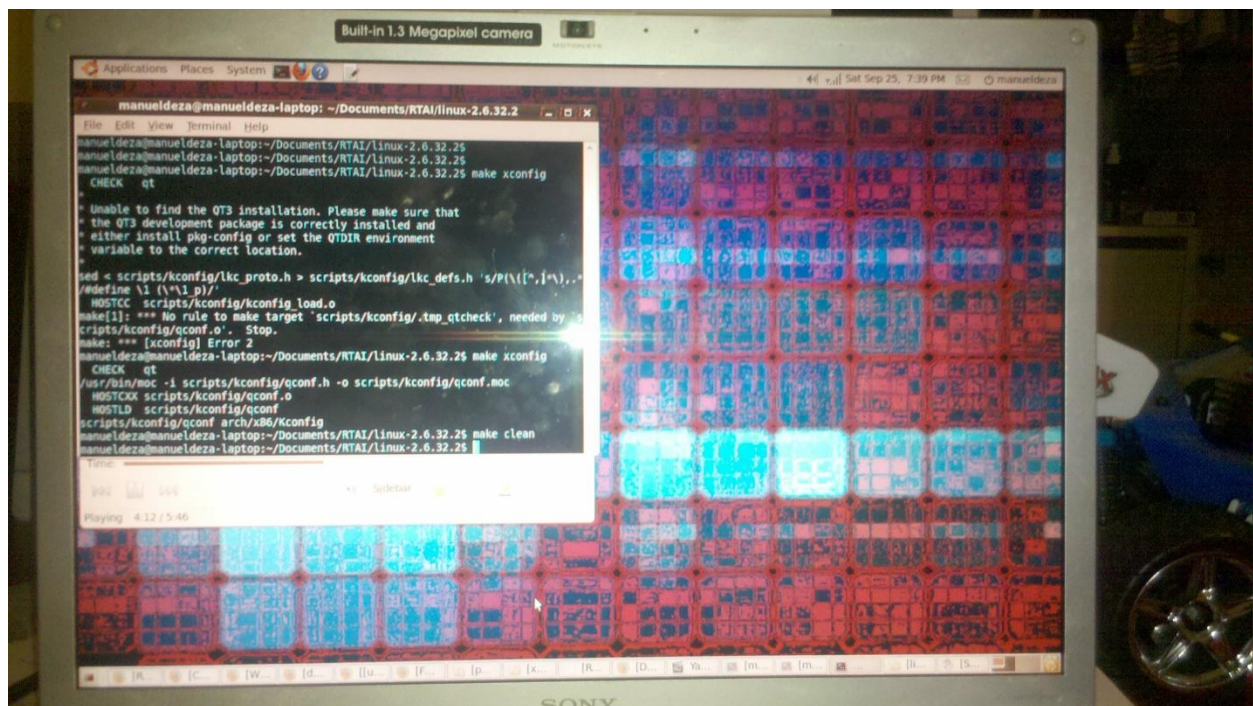
Open a terminal and type:





```
$ sudo su
#cd /usr/realtime/testsuite/kern/latency
#./run
```

```
File Edit View Terminal Help
manueldeza@manueldeza-laptop:/usr/realtime/testsuite/user/latencys s
[sudo] password for manueldeza:
*
*
* Type ^C to stop this application.
*
*
## RTAI latency calibration tool ##
# period = 1000000 (ns)
# average time = 1 (s)
# use the FPU
# start the timer
# timer_mode is oneshot
```





## **2: Doing some Speed tests with Rtai 3.8**

Program Code:

**Source Code: "third.c"**

```
#include <linux/init.h>
#include <linux/module.h>
#include <asm/current.h>
#include <linux/sched.h>
#include <linux/time.h>
#include <linux/jiffies.h>
static int __init jiffies_init(void)
{
    unsigned long j,z;
    j = z = 0;
    j = jiffies;
    z = HZ;
    int m=0;
    for(m=0;m<=50;m++)
    {
        printk(KERN_ALERT "Valor de Contador m : %lu\n",m);
        printk(KERN_ALERT "jiffies value is %lu\n",j);
        printk(KERN_ALERT "jiffies value in seconds %lu\n",(jiffies/HZ));
        printk(KERN_ALERT "HZ value is %lu\n",z);
    }
    return 0;
}

static void __exit jiffies_exit(void)
{
    printk(KERN_ALERT "Goodbye, world!\n");
}

module_init(jiffies_init);
module_exit(jiffies_exit);

MODULE_LICENSE("GPL");
```

**Makefile Code "Makefile":**

```
# MAKEFILE
```



```
TARGET = third
```

```
EXTRA_CFLAGS += -I/usr/realtime/include
```

```
CURRENT = $(shell uname -r)
```

```
KDIR = /lib/modules/$(CURRENT)/build
```

```
PWD = $(shell pwd)
```

```
obj-m := $(TARGET).o
```

```
all:
```

```
$(MAKE) -C $(KDIR) M=$(PWD) modules
```

Instructions:

Make a folder inside Documents: realtime/third

Inside this folder we'll save "third.c" and "Makefile".

Now, to put the program in Kernel Space, we'll need an ".ko" or Kernel Object file.

To do this, we type the following in a terminal:

```
$ sudo su
```

```
# make
```

Notice that the following files have been created in the /Documents/realtime/third directory.

Module.makers

Module.symvers

modules.order

third.ko (archivo que usaremos)

third.mod.c

third.mod.o

third.o

Now we use the **lsmod** command to see what Modules the Kernel is using, thus burning our RAM.

```
$ lsmod
```



Observe in the list that the “third “ module isn’t inserted.

We’ll have to load the “third.ko” module.

```
# insmod third.ko
```

To check out the running operations

```
/usr/var/messages
```

Or we can use:

```
# dmesg -c
```

Where we observe the execution time of every microinstruction done by the kernel.

This time has a 1 microsecond precision, and shows the absolute time in seconds, ever since the computer was booted.

Coming up, we made an Excel chart that shows the execution time difference from a Generic Linux Kernel and an RTAI Linux patched kernel:

RTAI 2.6.31.8 kernel DATA				Generic- 2.6.31-14 kernel DATA			
m	Value	Difference	us Dif	m	Value	Difference	us Dif
0	1714.121132	0.000006	6.00	0	211.723633	0.000022	22.00
1	1714.121138	0.000004	4.00	1	211.723655	0.000018	18.00
2	1714.121142	0.000004	4.00	2	211.723673	0.000017	17.00
3	1714.121146	0.000005	5.00	3	211.723690	0.000017	17.00
4	1714.121151	0.000004	4.00	4	211.723707	0.000017	17.00
5	1714.121155	0.000004	4.00	5	211.723724	0.000017	17.00
6	1714.121159	0.000004	4.00	6	211.723741	0.000016	16.00
7	1714.121163	0.000004	4.00	7	211.723757	0.000017	17.00
8	1714.121167	0.000004	4.00	8	211.723774	0.000017	17.00
9	1714.121171	0.000005	5.00	9	211.723791	0.000017	17.00
10	1714.121176	0.000004	4.00	10	211.723808	0.000017	17.00
11	1714.121180	0.000004	4.00	11	211.723825	0.000017	17.00
12	1714.121184	0.000004	4.00	12	211.723842	0.000017	17.00





13	1714.121188	0.000004	4.00	13	211.723859	0.000017	17.00
14	1714.121192	0.000005	5.00	14	211.723876	0.000017	17.00
15	1714.121197	0.000004	4.00	15	211.723893	0.000017	17.00
16	1714.121201	0.000004	4.00	16	211.723910	0.000016	16.00
17	1714.121205	0.000004	4.00	17	211.723926	0.000017	17.00
18	1714.121209	0.000004	4.00	18	211.723943	0.000017	17.00
19	1714.121213	0.000004	4.00	19	211.723960	0.000017	17.00
20	1714.121217	0.000004	4.00	20	211.723977	0.000018	18.00
21	1714.121221	0.000005	5.00	21	211.723995	0.000016	16.00
22	1714.121226	0.000004	4.00	22	211.724011	0.000017	17.00
23	1714.121230	0.000004	4.00	23	211.724028	0.000017	17.00
24	1714.121234	0.000004	4.00	24	211.724045	0.000018	18.00
25	1714.121238	0.000004	4.00	25	211.724063	0.000017	17.00
26	1714.121242	0.000004	4.00	26	211.724080	0.000016	16.00
27	1714.121246	0.000004	4.00	27	211.724096	0.000017	17.00
28	1714.121250	0.000004	4.00	28	211.724113	0.000017	17.00
29	1714.121254	0.000005	5.00	29	211.724130	0.000017	17.00
30	1714.121259	0.000004	4.00	30	211.724147	0.000017	17.00
31	1714.121263	0.000004	4.00	31	211.724164	0.000017	17.00
32	1714.121267	0.000004	4.00	32	211.724181	0.000017	17.00
33	1714.121271	0.000004	4.00	33	211.724198	0.000017	17.00
34	1714.121275	0.000004	4.00	34	211.724215	0.000017	17.00
35	1714.121279	0.000004	4.00	35	211.724232	0.000017	17.00
36	1714.121283	0.000005	5.00	36	211.724249	0.000016	16.00
37	1714.121288	0.000004	4.00	37	211.724265	0.000017	17.00
38	1714.121292	0.000004	4.00	38	211.724282	0.000017	17.00
39	1714.121296	0.000004	4.00	39	211.724299	0.000017	17.00
40	1714.121300	0.000004	4.00	40	211.724316	0.000017	17.00
41	1714.121304	0.000004	4.00	41	211.724333	0.000017	17.00
42	1714.121308	0.000005	5.00	42	211.724350	0.000017	17.00
43	1714.121313	0.000004	4.00	43	211.724367	0.000018	18.00
44	1714.121317	0.000004	4.00	44	211.724385	0.000017	17.00
45	1714.121321	0.000004	4.00	45	211.724402	0.000017	17.00
46	1714.121325	0.000004	4.00	46	211.724419	0.000018	18.00
47	1714.121329	0.000004	4.00	47	211.724437	0.000017	17.00
48	1714.121333	0.000004	4.00	48	211.724454	0.000018	18.00
49	1714.121337	0.000004	4.00	49	211.724472	0.000017	17.00
50	1714.121341			50	211.724489		

**Time difference for every microinstruction process varies significantly. Even though inperceptual, we are talking about an (x4) processing acceleration.**

2.6.31.8 Kernel patched with Rtai 3.8 had userspace microprocesses that lasted from 4 to 6 microseconds for every 4 operations.

Generic Kernel had about 16 to 22 microseconds per every 4 operations.



### **3: Common Bugs \***

Here are some photos of common error screens, during my first attempts to install RTAI.

```
nake131: Leaving directory '/opt/magna/addons'
nake121: Leaving directory '/opt/magna/addons'
nake111: Leaving directory '/opt/magna/addons'
Making install in rtai-py
nake111: Entering directory '/opt/magna/rtai-py'
nake121: Entering directory '/opt/magna/rtai-py'
/bin/bash ../base/config/autoconf/nkinstalldirs /usr/realtime/rtai-py
mkdir -p -- /usr/realtime/rtai-py
nake121: Nothing to be done for 'install-data-am'.
nake121: Leaving directory '/opt/magna/rtai-py'
nake111: Leaving directory '/opt/magna/rtai-py'
nake111: Entering directory '/opt/magna'
nake121: Entering directory '/opt/magna'
grep: /etc/udev/rules.d/99-rtai.rules: No such file or directory
/bin/bash ../base/config/autoconf/nkinstalldirs /usr/realtime/share/rtai
mkdir -p -- /usr/realtime/share/rtai
/usr/bin/install -c -m 644 .rtai_config /usr/realtime/share/rtai/config-rtai-mag
na
if test -f Module.synvers ; then \
    /usr/bin/install -c -m 644 Module.synvers /usr/realtime/modules/Modu
le.synvers ; \
fi
nake121: Leaving directory '/opt/magna'
nake111: Leaving directory '/opt/magna'
root@manueldeza-laptop:/opt/magna#
```

Xorg.conf Error (X-server) :

## “Mounting device, sys and proc Boot error”

11





```
I 0.0179971 ..MP-BIOS bug: 8254 timer not connected to IO-APIC
modprobe: FATAL: Could not load /lib/modules/2.6.32.2/modules.dep: No such file
or directory

mount: mounting /dev/disk/by-uuid/c1048166-ced9-4389-be87-e2c384c04232 on /root
failed: No such device
mount: mounting /sys on /root/sys failed: No such file or directory
mount: mounting /dev on /root/dev failed: No such file or directory
mount: mounting /sys on /root/sys failed: No such file or directory
mount: mounting /proc on /root/proc failed: No such file or directory
Target filesystem doesn't have /sbin/init.
No init found. Try passing init= bootarg.

BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu7) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
(initramfs) _
```

Reminder: The best way to solve the bugs, is to google them up.



## 4: Additional Results:

Monitoring my progress:



**91 visits came from 14 countries/territories**

Detail Level: [City](#) | [Country/Territory](#) | [Sub Continent Region](#) | [Continent](#) Dimension: [None](#)

Using Google Analytics, I found out that from November – December :42 of 91 visits of my webpage were from Italy, country where RTAI was developed. Hooray!

\*RTAI was developed at Politécnico di Milano, Italy.

Site Usage		Goal Set 1
<b>Visits</b> <b>91</b> % of Site Total: 100.00%		<b>Pages/Visit</b> <b>1.48</b> Site Avg: 1.48 (0.00%)
		<b>Avg. Time on Site</b> <b>00:02:11</b> Site Avg: 00:02:11 (0.00%)
Detail Level: <a href="#">Country/Territory</a>		Visits ↓
1.	<a href="#">Italy</a>	42
2.	<a href="#">United States</a>	11
3.	<a href="#">Peru</a>	11

I'm happy I could give an extra hand to the RTAI community!



## **5: Bibliographic materials:**

Most guides and bibliographic material at the time this guide and report was made were done for previous versions of Ubuntu Linux, such as 8.04 and 8.10, as well as previous versions of RTAI (3.3 and 3.5). Today, we are at Ubuntu version 10.10, and one of the latest kernels is 2.6.35-22-generic. The challenge for making this guide was to make it compatible and working for the newest available versions of both Ubuntu and RTAI. I was surprised that most people out there when wanting to work with RTAI would just lower their Ubuntu version as a solution, but eventually time will pass, and new software's evolve: one must always try to be up to date.

### **Other RTAI Installation Guides:**

A guide to installing RTAI Linux – Keith ShortRidge -2004

RTAI Installacion Complete Guide – Joao Monteiro -2008

How to install RTAI into Ubuntu Hardy – Cristovao Sousa -2008

### **Kernel Space Developing:**

Introduction to Linux Device Driver Development – Richard Sevenich - 2004

Writing Linux Kernel Modules – Kernel Fibonacci Series Generator – Muli Ben-Yehuda -2003

Loadable Kernel Modules – Juan-Mariano de Goyeneche & Elena Apolinario Fernandez de Sousa – 1999

### **RTAI Programming:**

RTAI API Base – Antonio Barbalace – 2009

RTAI 3.3 User Manual

Building your way through RTAI – Joao Monteiro -2008





## **6: Final Note for the author:**

Please if you have any suggestions, or found any errors, don't hesitate on emailing me. I would gladly help anyone out there.

Email me at: mdezaf [at] uni .pe